

Roues codeuses RC
(Adressage)

Déci
0-9

Hexa
0-F

EEPROM

Jumper Activation 2^{ème} balise

Switch « Action Events »

Inter « Failure Mode » ON/OFF

Paramètres Cfgs du
Dipswitch

Debug
D 1-2
EE
FEUX
ConfA
ConfB
Fail1
Fail2

- mc_InfraBAL_parametreCarte (Only Remote)
- mc_InfraBAL_statusDynamique (Only Remote)
- mc_InfraBAL_feu_A (events)
- mc_InfraBAL_feu_B (events)
- mc_InfraBAL_EEPROM (events Ecriture)
- Trames étendues pour réception message EBT
- mc_InfraBAL_sliceEbtL1_ToBalA (events)
- mc_InfraBAL_sliceEbtL1_ToBalA+1 (events)
- mc_InfraBAL_sliceEbtL1_ToBalA+2 (events)
- mc_InfraBAL_sliceEbtL1_ToBalB (events)
- mc_InfraBAL_sliceEbtL1_ToBalB+1 (events)
- mc_InfraBAL_sliceEbtL1_ToBalB+2 (events)

- mc_InfraBAL_parametreCarte (Emission si Remote)
- mc_InfraBAL_statusDynamique (Emission si Remote)
- mc_InfraBAL_CRO_EbtL2_BalA (events)
- mc_InfraBAL_CRO_EbtL2_BalB (events)
- mc_InfraBAL_capteurOnCI_A (events)
- mc_InfraBAL_capteurOnCI_B (events)
- mc_InfraBAL_feu_A (CRO events)
- mc_InfraBAL_feu_B (CRO events)
- mc_InfraBAL_DV_A (events)
- mc_InfraBAL_DV_B (events)
- mc_InfraBAL_EEPROM (events Lecture)
- Communes à l'ensemble des cartes :
- mc_Infra_crash (events)
- mc_Infra_warning (events)
- mc_alimentationRails (events)
- mc_ledRouge_On_Tour

Carte Infra Balise

DSPIC Infra Balise

Led Status balise A
-Verte: OK
-Rouge: Bad

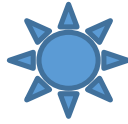
Led Status balise B
-Verte: OK
-Rouge: Bad

Balise A
Communication Infra-Rouge vers locomotive
Capteur Hall Détection Loco dessus Balise A

Balise B
Communication Infra-Rouge vers locomotive
Capteur Hall Détection Loco dessus Balise B



Feu
BaliA



Feu
BaliB



Capteur de
Voie
DV_BalA



Capteur de
Voie
DV_BalB

mc_InfraBAL_parametreCarte

ID	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
(RC << 4) +0x1 : Carte Balise	6	xx	xx	xx	xx	xx	xx		



I/O

Les cartes Balises sont équipées de Roues Codeuses, de Dipswitch, d'un cavalier et d'un inter. Ils servent à paramétrer les fonctions de chaque carte. Cette trame CAN n'est pas envoyée spontanément. Elle sera uniquement émise après une trame de remote.

Ces valeurs sont en principe lues une seule fois en début de code ou à la mise sous tension de la carte. **Une trame remote provoquera une nouvelle lecture des bits suivants** qui peuvent avoir une signification. (configuration dynamique de la carte)

- bdmc_InfraBAL_modeProtectEEPROM
- bdmc_InfraBAL_fail1 (En attente de codage)
- bdmc_InfraBAL_fail2 (En attente de codage)
- bdmc_InfraBAL_activationModeFailure (Interrupteur Mode Failure ON/OFF placé au centre de la carte balise) (En attente de codage)
- bdmc_InfraBAL_switchActionEvent (switch « action/Events placé au dessus de l'interrupteur Mode Failure ON/OFF)

En effet, les autres bits ne doivent pas changer, ils sont liés à la configuration Hardware de la carte Balise.

D[0] = cdmc_InfraBAL_RC (codage hexadécimal des 2 roues codeuses)

D[1] = cdmc_InfraBAL_cfgs_Dipswitch

- bit7: bdmc_InfraBAL_modeDebug (High -> mode debug activé) (En attente de codage)
- bit6: bdmc_InfraBAL_modeDebug1or2 (low -> mode 2 activé, high -> mode 1 activé) (En attente de codage)
- bit5: bdmc_InfraBAL_modeProtectEEPROM (Low -> read/write EEPROM, High -> read only)
- bit4: bdmc_associeBaliseEtFeux (Low -> Aucun des 2 feux de la carte n'est actif, High -> dépend du bit ci-dessous)
- bit3: bdmc_confA_BaliseFeux (Low -> Seule la balise A est associée à un feu, High -> Les balises A et B sont chacune associées à un feu)
- PS1: La configuration (bdmc_associeBaliseEtFeux = Low) ET (bdmc_confA_BaliseFeux = High) EST INTERDITE !!!!!!!
- PS2: Bit4 devient « Don't care » si « bdmc_InfraBAL_activationBalB » = Low.
- bit2: bdmc_confB_GroupeBalises (En attente de codage)
- bit1: bdmc_InfraBAL_fail1 (High -> Pour provoquer pannes sur cartes) (En attente de codage)
- bit0: bdmc_InfraBAL_fail2 (High -> Pour provoquer pannes sur cartes) (En attente de codage)

D[2] = cdmc_InfraBAL_confSupp

- bit0: bdmc_InfraBAL_activationBalB (Low -> No, High -> Yes). Etat du jumper « 2 ième balise » (N'est lu qu'au démarrage programme)
- bit1: bdmc_InfraBAL_activationModeFailure (Low -> No, High -> Yes). Etat de l'inter « Failure Mode »
- bit2: bdmc_InfraBAL_switchActionEvent (High -> Appui sur switch) En mode normal, provoquera l'extinction du bargraph et variables associées. Aura aussi la même fonction que l'envoi de la remote trame mc_statusDynamiqueCarteBalise avec un DLC de 0.

D[3] = cdmc_numero_BalA

D[4] = cdmc_numero_BalB

D[5] = cdmc_InfraBAL_VersionSoft

mc_InfraBAL_statusDynamique

ID	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
(RC << 4) + 0xD : Carte Balise	7	xx	xx	xx	xx	xx	xx	xx	xx

↑ OUT

D[0] = cdmc_etatLedsBargraph (L'état de ces bits est recopié automatiquement sur le bargraph.)

bit7: bdmc_InfraBAL_CAN_TxErrorPassive

(Led rouge)

bit6: bdmc_InfraBAL_CAN_RxErrorPassive

(Led rouge)

bit5: bdmc_warningFailureCommBalA

(Led orange)

bit4: bdmc_warningFailureCommBalB

(Led orange)

bit3: bdmc_InfraBAL_defEEPROM

(Led orange)

bit2: bdmc_InfraBAL_def1TrameErronee

(Led orange)

bit1: bdmc_InfraBAL_def2

(Temporaire, bug soft « mise au point programme »)

(Led rouge)

bit0: bdmc_InfraBAL_def3

(Temporaire,Bug soft avec variable erreur librairie CAN)

(Led rouge)

D[1] = cdmc_cptBadComIR_WithBalA

(Compteur indiquant le nombre d'échecs de communication IR intervenu avec la balise A)

D[2] = cdmc_cptBadComIR_WithBalB

(Compteur indiquant le nombre d'échecs de communication IR intervenu avec la balise B)

D[3] = cdmc_InfraBAL_cptTrameErronee

D[4] = cdmc_InfraBAL_cptTxCAN_Error

(Copie du cpt « Périphérique CAN)

D[5] = cdmc_InfraBAL_cptRxCAN_Error

(Copie du cpt « Périphérique CAN)

D[6],D[7] = wdmc_InfraBAL_ID_TrameErronee

(Identifiant CAN codé sur 2 octets de la dernière trame erronée reçue).

PS: Si l'identifiant est « étendu », on ne garde que les 16 bits de poids faibles, les poids forts étant toujours à « 0 »

Cette trame ne sera jamais émise spontanément par une carte balise, mais uniquement en réponse à une remote trame. Une trame de datas ne sera pas prise en compte.

- Au niveau « Remote trame », deux cas possibles:
- A) Une « remote trame » avec un DLC de 0.
 - B) Une « remote trame » avec un DLC différent de 0.
- Chacune de ces trames renverra les datas de la trame « **mc_InfraBAL_statusDynamique** », mais, en plus, celle avec le DLC=0 resettera tous les bits et compteurs d'erreurs, elle éteindra aussi les 2 leds bicolores « status Balise A et B)
- PS: « cdmc_InfraBAL_cptTxCAN_Error » et « cdmc_InfraBAL_cptRxCAN_Error » qui sont des images du registre d'erreurs du CAN ne peuvent être remis a 0.
- PS1: Une remote trame avec un DLC différent de 0 ne change aucune data.
- PS2: Rappel, l'appui sur le swicht « Events » situé sur la carte balise a la même fonction que cette « remote trame » avec son DLC à 0.

Message Eurobalise Telegram (Ebt)

mc_InfraBAL_sliceEbtL1_ToBalA ou mc_InfraBAL_sliceEbtL1_ToBalB

EID (Extend ID)	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
(RC << 8) +0x00 : BaliseA	8	len_EbtL0_BalA	CKS_EbtL1_BalA	standardCom_BalA	dataSup_BalA	Reserved 0x00	Reserved 0x00	Reserved 0x00	Reserved 0x00
(RC << 8) +0x80 : BaliseB	8	len_EbtL0_BalB	CKS_EbtL1_BalB	standardCom_BalB	dataSup_BalB	Reserved 0x00	Reserved 0x00	Reserved 0x00	Reserved 0x00



IN

La description du message Ebt est codé dans l'identifiant étendu de la trame CAN. Les datas « Ebt » sont insérées dans les trames CAN contiguës et suivantes avec un maximum de 8 datas par trame. On répète l'opération jusqu'à la fin des datas Ebt. L : Level
PS: Le DLC de la dernière trame est donc variable suivant le nombre de datas du Ebt.

- D[0] : cdmc_len_EbtL0_BalA, cdmc_len_EbtL0_BalB: Ce nombre équivaut aussi à la somme des DLC de toutes les trames Datas Ebt à partir de l'Id étendu [(RC << 8)+0x01] pour la destination Balise A et [(RC << 8) +0x81] pour la destination Balise B.
- D[1] : cdmc_CKS_EbtL1_BalA, cdmc_CKS_EbtL1_BalB: Somme des datas[2] à data[7] pour l'id = [(RC << 8) +0x00] (si balise A) additionné de la somme des data[i] des « Id » supplémentaires et nécessaires. (Dépend de la longueur du message Ebt). Même logique que ci-dessus pour la balise B.
- D[2] : cdmc_standardCom_BalA, cdmc_standardCom_BalB: Information supplémentaire pour loco. Prévue pour indiquer le protocole (ERTMS, national, etc.)
- D[3] : cdmc_dataSup_BalA, cdmc_dataSup_BalB: Information supplémentaire pour loco.

EID (Extend ID)	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
(RC << 8) +0x01 : BaliseA (RC << 8) +0x81 : BaliseB	Var	EbtL0[0]	EbtL0[1]	EbtL0[2]	EbtL0[3]	EbtL0[4]	EbtL0[5]	EbtL0[6]	EbtL0[7]

EID (Extend ID)	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
(RC << 8) +0x02 : BaliseA (RC << 8) +0x82 : BaliseB	Var	EbtL0[8]	EbtL0[9]	EbtL0[10]	EbtL0[11]	EbtL0[12]

Compte Rendu Opérationnel du transfert de l'Eurobalise Telegram (Ebt) vers la loco via une balise

DSPIC Infra Balise

Mode opératoire d'envoi de ces trames: Quand une loco est passée sur une carte de communication Infra-Rouge (Connectée à une balise A ou une balise B), des données sont échangées entre cette carte et la loco. En particulier, la carte Infra IR transmet un Euro balise telegram 'Ebt' à la loco. Deux cas possibles:

A) La communication est OK, la trame « mc_InfraBAL_CRO_EbtL2_BalX » est émise, La data[2] soit « cdmc_InfraBAL_CRO_transfert_EbtL2_BalX » indique tous les possibles dysfonctionnements qui auraient pu entraver la communication. (PS, RAS si 0x00). Quelque soit cet octet, le transfert s'est bien déroulé, le code a tout « Rattrapé ».

B) La communication a échouée, dans ce cas c'est une trame CAN « Warning » qui est émise.

PS: Quelques datas supplémentaires sont aussi transmises à la loco telle que le standard de communication, une data supplémentaire, le numéro de la balise émettrice de l'Ebt et l'état du feu suivant. Tout ceci de façon transparente pour l'utilisateur.

En résumé, quand une trame « mc_InfraBAL_CRO_EbtL2_BalX » est émise, la communication s'est bien déroulée, le transfert des datas est OK.

mc_InfraBAL_CRO_EbtL2_BalA

↑ OUT

SID (Standard ID)	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
(RC << 4) + 0x2 : BaliseA	2	xx	xx						

D[0] : cdmc_InfraBAL_numeroLoco_BalA :

D[1] : cdmc_InfraBAL_CRO_transfert_EbtL2_BalA (si 0x00, aucun problème mineur n'est survenu, si !=0, les datas de cette trame sont OK.

bits[3..0]: Compteur (écrête a 0x0F) de tous les dysfonctionnements intervenus

bit4: Erreur CKS

bit5: Erreur TimeOUT

mc_InfraBAL_CRO_EbtL2_BalB

↑ OUT

SID (Standard ID)	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
(RC << 4) + 0xA : BaliseB	2	xx	xx						

D[0] : cdmc_InfraBAL_numeroLoco_BalB:

D[1] : cdmc_InfraBAL_CRO_transfert_EbtL2_BalB

bits[3..0]: Compteur (écrête a 0x0F) de tous les dysfonctionnements intervenus

bit4: Erreur CKS

bit5: Erreur TimeOUT

Trames CAN envoyées lors des détections locos au dessus de la balise A ou de la balise B

Mode opératoire d’envoi de ces trames: Lorsqu’un capteur magnétique situé sur une carte communication Infra Rouge (sous la voie) détecte une locomotive, une trame CAN est transmise. Aucune trame ne sera transmise lors du relâchement du capteur. Une nouvelle lecture de ce capteur aura lieu une seconde après, si la loco est toujours présente, une autre trame sera transmise et ceci, indéfiniment.

mc_InfraBAL_capteurOnCI_BalA, mc_InfraBAL_capteurOnCI_BalB

↑
OUT

ID	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
(RC << 4) +0x3 : BaliseA	3	0x44	xx	xx					
(RC << 4) +0xB : BaliseB	3	0x44	xx	xx					

D[0] : cdmc_typeData_CapteurOnCI_BalA, cdmc_typeData_CapteurOnCI_BalB
D[1] : cdmc_etat_CapteurOnCI_BalA, cdmc_etat_CapteurOnCI_BalB
bit0: Détection capteur à effet Hall, (gauche ou droite)
bit1: Détection capteur à effet Hall, (l'autre)
bit4: Détection position loco par rapport à cette carte.

PS: Tous ces bits sont indépendants.
D[2] : cdmc_status_CapteurOnCI_BalA, cdmc_status_CapteurOnCI_BalAB
0x00 :

0x01 :

toujours à 'D' pour « Data » soit (Code ASCII de D = 0x44)

Dépend du sens de montage de la carte

Le transducteur utilisé pour la communication Infrarouge est dissymétrique. A cause de cette particularité, on peut avoir la carte balise sous rail qui reçoive parfaitement le signal envoyé par la loco, Que celle-ci renvoie un ACK mais que cet ACK soit non reçu par la loco. Ceci ne se produit que dans un seul sens de circulation, et lorsque la loco est décalée d'environ 7 cm axe à axe capteur IR.

En réponse à une remote trame, ou lors d'une première détection de la présence loco dessus le capteur
La loco n'a pas quitté le capteur. Chaque seconde après la présence loco dessus le capteur, si la loco est toujours présente dessus le capteur, une trame est renvoyée avec cette valeur.

Trames CAN pour la gestion des feux connectés à la carte balise

On distingue deux types de trames:

- Une trame de positionnement d'une couleur à un feu. Un dispositif envoie une trame CAN à la carte « Actionneur » dont la D[0]='W' et dont D[1] est la couleur désirée.
- Une trame de compte rendu opérationnel renvoyée par la carte « Actionneur ». Dans ce cas D[0] = 'D', D[1] est la couleur du feu, et D[3] est le CRO. Cette même trame est aussi renvoyée en réponse à une « remote trame ».

Rappel: Suivant l'état des switch, la carte peut piloter 0,1 ou 2 feux.

mc_InfraBAL_feu_A, mc_InfraBAL_feu_B

I/O	ID	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
	(RC << 4) + 0x4 : Feu A (RC << 4) + 0x5 : Feu B	3	xx	xx	xx					

D[0] : cdmc_InfraBAL_typeData_Feu_A ou cdmc_InfraBAL_typeData_Feu_B

'W' ou 0x57 : Trame d'action envoyée par un autre dispositif pour le positionnement d'une couleur à un feu.

'D' ou 0x44 : Trame de datas renvoyée par la carte Balise (Compte rendu opérationnel ou réponse à une « Remote trame »).

D[1] : cdmc_InfraBAL_couleur_Feu_A ou cdmc_InfraBAL_couleur_Feu_B

0x00 : Feu éteint

0x01 : Feu rouge

0x02 : Feu vert

0x03 : Feu rouge clignotant

0x04 : Feu vert clignotant

0x05 : Feu orange

D[2] : cdmc_InfraBAL_status_Feu_A ou cdmc_InfraBAL_statusFeu_B

0x00 : RAS

0x01 : MODULE_DEJA_DANS_ETAT_DEMANDE, Tout est OK, sauf que le feu était déjà dans la couleur demandée.

0x02 : La trame est erronée. Soit une erreur de DLC ou de datas non prévues

Trames CAN pour la gestion des détecteurs de voie

Mode opératoire d’envoi de ces trames: Lorsqu’un capteur magnétique ou détecteur de voie détecte une locomotive, une trame CAN est transmise. Aucune trame ne sera transmise lors du relâchement du capteur. Une nouvelle lecture de ce capteur aura lieu une seconde après, si la loco est toujours présente, une autre trame sera transmise et ceci, indéfiniment.

mc_InfraBAL_DV_A, mc_InfraBAL_DV_B

↑ Out	ID	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
	(RC << 4) +0x6 : DV A (RC << 4) +0x7 : DV B	3	xx	xx	xx					

- D[0] : cdmc_typeData_DV_A ou cdmc_typeData_DV_B.
‘D’ ou 0x44 : Trame de datas qui indique l’état du capteur.

Pas de remote trame de prévue. Cette data n’est pas significative, c’est uniquement pour garder une certaine compatibilité avec des autres trames similaires.
- D[1] : cdmc_etat_DV_A ou cdmc_etat_DV_B
0x01 : Présence loco au dessus du capteur
- D[2] : cdmc_status_DV_A ou cdmc_statusDV_B
0x00 : En réponse à une remote trame, ou lors d’une première détection de la présence loco dessus le capteur
0x01 : La loco n’a pas quitté le capteur.

Chaque seconde après la présence loco dessus le capteur, si la loco est toujours présente dessus le capteur, une trame est renvoyée avec cette valeur.

Trames CAN pour la mémoire EEPROM de la carte Balise

mc_InfraBAL_EEPROM



ID	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
(RC << 4) + 0xE	x	xx	xx	xx	xx	xx	xx	xx	xx

D[0] : cdmc_InfraBAL_commande_EEPROM

‘O’ (Code ASCII de O = 0x4F) -> Paramétrage OFFSET

‘W’ (Code ASCII de W = 0x57) -> Ecriture EEPROM

‘D’ (Code ASCII de D = 0x44) -> Lecture EEPROM (n’est pas une commande, est uniquement renvoyé après une remote trame).

D[1] : cdmc_InfraBAL_data1_EEPROM

D[2] : cdmc_InfraBAL_data2_EEPROM

ou

D[1], D[2] : wdmc_InfraBAL_offset_EEPROM si D[0]=‘O’

D[3] : cdmc_InfraBAL_data3_EEPROM

D[4] : cdmc_InfraBAL_data4_EEPROM

D[5] : cdmc_InfraBAL_data5_EEPROM

D[6] : cdmc_InfraBAL_data6_EEPROM

D[7] : cdmc_InfraBAL_data7_EEPROM

NB: Pas d’auto-indentation de « wdmc_InfraBAL_offset_EEPROM »

ATTENTION, L’EEPROM peut être protégée en écriture à l’aide du Dipswitch sur la carte. Il suffit d’activer le switch EE->U vers la droite.

Si on tente une écriture dans l’EEPROM, un message warning est envoyé sur le bus avec les paramètres suivants:

Id warning = 0x0D : Trame dysfonctionnement survenu sur l’infrastructure.

D[0] -> Type de la carte qui a envoyé le message warning (0x01 -> carte balise)

D[1] -> Type warning, ici 0x10 pour type SOFTWARE

D[2] -> Valeur de la roue codeuse de la carte

D[3] -> Type du Warning soit 0xEE dans ce cas

PS: La led EEPROM du Bargraph s’allume aussi.

Mode d’emploi EEPROM:

- Paramétrage de l’adresse d’offset (pointeur):

Le PC envoie une trame standard avec D[0] = ‘O’ (Code ASCII de O = 0x4F) suivi de la valeur de l’offset codé de la façon suivante:

D[2] pour poids forts et D[1] pour poids faibles

OFFSET = (unsigned int) ((D[2] << 8) + D[1]) codage “Loco”

- Écriture dans l’EEPROM :

Le PC envoie une trame standard avec D[0] = ‘W’ (Code ASCII de W = 0x57) suivi des valeurs à écrire dans l’EEPROM. Le DLC indique le nombre de valeurs à écrire.

À l’issue de l’écriture, la carte envoie un « ACK » pour signaler la fin de l’opération. Pour ce faire, une trame est renvoyée avec un DLC=1 et une D[0] = ‘A’ (0x41) pour ACK.

Si l’EEPROM était en mode « Read Only » (protection en écriture à l’aide du switch3 « bdmc_InfraBAL_modeProtectEEPROM » de la carte balise), la carte renverrait une trame « warning ».

PS: Si D[0] n’est ni ‘O’ ni ‘W’, la variable « cdmc_InfraBAL_cptTrameErronee » du message « mc_InfraBAL_statusDynamique » sera incrémentée..

De même que le bit « bdmc_InfraBAL_def1TrameErronee » de cette trame.

Par contre, le bit « bdmc_InfraBAL_defEEPROM » reste inchangé, on n’a pas eu d’erreur véritable au niveau de l’EEPROM.

- Lecture de l’EEPROM :

Le PC envoie une trame de Remote avec l’Id EEPROM

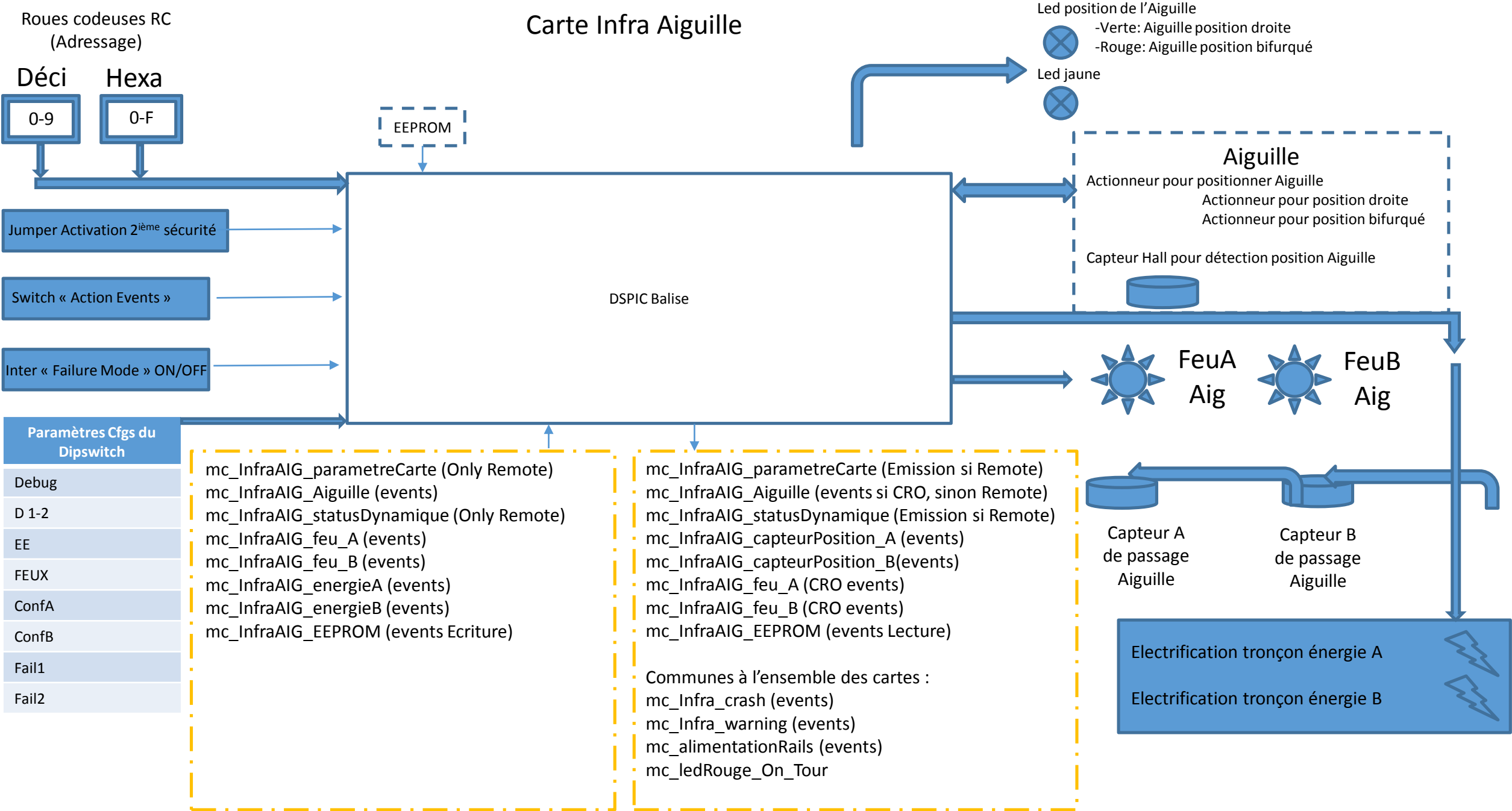
Le DLC de la trame remote indiquera le nombre de datas à renvoyer

La carte renverra une trame avec

D[0] = ‘D’ (Code ASCII de D = 0x44), suivi du nombre de datas demandés.

NB: Le DLC max de la trame de remote est de 7, car D[0] = ‘D’, reste donc 7 octets de disponibles dans la trame de renvoi.

PS: Si DLC=8, alors les variables « cdmc_InfraBAL_cptTrameErronee » et « bdmc_InfraBAL_def1TrameErronee » seront mis à jour.



mc_InfraAIG_parametreCarte

ID	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
(RC << 4) + 0x1 : Carte Aiguille	6	xx	xx	xx	xx	xx	xx		



I/O

Les cartes « Aiguille » sont équipées de Roues Codeuses, de Dipswitch, d'un cavalier et d'un inter. Ils servent à paramétrer les fonctions de chaque carte. Cette trame CAN n'est pas envoyée spontanément. Elle sera uniquement émise après une trame de remote.

Ces valeurs sont en principe lues une seule fois en début de code ou à la mise sous tension de la carte. Une trame remote provoquera une nouvelle lecture des bits suivants qui peuvent avoir une signification. (configuration dynamique de la carte)

bdmc_InfraAIG_modeProtect_EEPROM

bdmc_InfraAIG_fail2 (En attente de codage)

bdmc_InfraAIG_fail2 (En attente de codage)

bdmc_InfraAIG_activationModeFailure (Interrupteur Mode Failure ON/OFF placé au centre de la carte balise) (En attente de codage)

bdmc_InfraAIG_switchActionEvent. (switch « action/Events placé au dessus de l'interrupteur Mode Failure ON/OFF)

En effet, les autres bits ne doivent pas changer, ils sont liés à la configuration Hardware de la carte Balise.

D[0] = cdmc_InfraAIG_RC

(codage hexadécimal des 2 roues codeuses)

D[1] = cdmc_InfraAIG_cfgs_Dipswitch

bit7: bdmc_InfraAIG_modeDebug

(High -> mode debug activé) (En attente de codage)

bit6: bdmc_InfraAIG_modeDebug_1or2

(low -> mode 2 activé, high -> mode 1 activé) (En attente de codage)

bit5: bdmc_InfraAIG_modeProtect_EEPROM

(Low -> read/write EEPROM, High -> read only)

bit4: bdmc_associeAiguilleEtFeux

(Low -> Aucun des 2 feux de la carte n'est actif, High -> dépend du bit ci-dessous)

bit3: bdmc_confA_NombreFeuxActifs

(Low -> 1 seul feu actif, High -> Les 2 feux sont actifs)

PS1: La configuration (bdmc_associeAiguilleEtFeux = Low) ET (bdmc_confA_NombreFeuxActifs = High) EST INTERDITE !!!!!!!

bit2: bdmc_confB_NombreHallActifs

(Low -> un seul capteur de détection est activé, High -> les deux capteurs sont activés)

bit1: bdmc_InfraAIG_fail1

(En attente de codage)

bit0: bdmc_InfraAIG_fail2

(En attente de codage)

D[2] = cdmc_InfraAIG_confSupp

bit0: bdmc_InfraAIG_activationSecu

(Low -> No, High -> Yes). Etat du jumper « 2 ième sécurité » (N'est lu qu'au démarrage programme) En attente de codage!

bit1: bdmc_InfraAIG_activationModeFailure

(Low -> No, High -> Yes). Etat de l'inter « Failure Mode » En attente de codage!

bit2: bdmc_InfraAIG_switchActionEvent

(High -> Appui sur switch) En mode normal, provoquera l'extinction du bargraph et variables associées. Aura aussi la même fonction que l'envoi de la remote trame mc_InfraBAL_statusDynamique avec un DLC de 0.

D[3] = cdmc_numero_energieA

(Numéro de Tronçon ou d'aiguillages alimenté en énergie pour le déplacement du train)

D[4] = cdmc_numero_energieB

(Numéro de Tronçon ou d'aiguillages alimenté en énergie pour le déplacement du train)

D[5] = cdmc_InfraAIG_VersionSoft

mc_InfraAIG_statusDynamique

ID	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
(RC << 4) + 0xD : Carte Aiguille	7	xx	xx	xx	xx	xx	xx	xx	xx

↑ OUT D[0] = cdmc_InfraAIG_etatLedsBargraph (L'état de ces bits est recopié automatiquement sur le bargraph.)

- bit7: bdmc_InfraAIG_CAN_TxErrorPassive (Led rouge)
- bit6: bdmc_InfraAIG_CAN_RxErrorPassive (Led rouge)
- bit5: bdmc_InfraAIG_Gripper (Led orange) Basculement dur dur !
- bit4: bdmc_InfraAIG_secu (Led orange) En attente de codage
- bit3: bdmc_InfraAIG_defEEPROM (Led orange)
- bit2: bdmc_InfraAIG_def1TrameErronee (Led orange)
- bit1: bdmc_InfraAIG_def2 (Led rouge) (Temporaire, bug soft « mise au point programme »)
- bit0: bdmc_panneAig (Led rouge)

D[1] = Reserved (0x00)

D[2] = cdmc_cptBascul_Aig_complique (Compteur indiquant le nombre de basculements dur dur de l'aiguille)

D[3] = cdmc_InfraAIG_cptTrameErronee

D[4] = cdmc_InfraAIG_cptTxCAN_Error (Copie du cpt « Périphérique CAN)

D[5] = cdmc_InfraAIG_cptRxCAN_Error (Copie du cpt « Périphérique CAN)

D[6] = cdmc_InfraAIG_PoidsFaibles_ID_TrameErronee (Poids faibles (8 bits d'offset) de l'Id de la dernière trame erronée reçue.

D[7] = cdmc_InfraAIG_PoidsForts_ID_TrameErronee (Poids forts (8 bits d'offset) de l'Id de la dernière trame erronée reçue.

PS: La led jaune s'allumera quand la var « cptErreurDefinition » sera différente de zéro (Debug prog)

Cette trame ne sera jamais émise spontanément par une carte Aiguille, mais uniquement en réponse à une remote trame. Une trame de datas ne sera pas prise en compte.

Au niveau « Remote trame », deux cas possibles:

- Une « remote trame » avec un DLC de 0.
- Une « remote trame » avec un DLC différent de 0.

Chacune de ces trames renverra les datas de la trame « mc_InfraAIG_statusDynamique », mais, en plus, celle avec le DLC=0 resettera tous les bits et compteurs d'erreurs, elle éteindra aussi la led bicolore et la jaune situées sur la carte.

PS1: D[1] ou « cdmc_InfraAIG_status_Aig » n'étant pas un indicateur d'erreur restera toujours inchangé

PS2: « cdmc_InfraAIG_cptTxCAN_Error » et « cdmc_InfraAIG_cptRxCAN_Error » qui sont des images du registre d'erreurs du CAN ne peuvent être remis à 0.

PS3: Une remote trame avec un DLC différent de 0 ne change aucune data.

PS4: Rappel, l'appui sur le swicht « Events » situé sur la carte balise a la même fonction que cette « remote trame » avec son DLC à 0.

On distingue deux types de trames CAN avec le même identificateur.

- Trame de positionnement de l'aiguille dans une position donnée. Cette trame est envoyée pour permettre le positionnement de l'aiguille. (Droit ou bifurqué)
- Trame de compte rendu opérationnel. Cette trame est renvoyée automatiquement à la suite de la commande précédente pour signifier le résultat du déroulement de l'opération. Cette trame est également renvoyée en réponse à une remote trame.

mc_InfraAIG_Aiguille

I/O

SID (Standard ID)	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
(RC << 4) + 0x2 : Aiguille	3	xx	xx	xx					

D[0] : cdmc_InfraAIG_typeData:

D[1] : cdmc_InfraAIG_position

D[2] : cdmc_InfraAIG_lock_or_CRO

Datas en Mode « Set position aiguille »

D[0] : On distingue 4 types de commande possibles:

D[0] = 'W' ou 0x57 : Positionnement Aig

D[0] = 'L' ou 0x4C : Verrouillage ou Lock Aig

D[0] = 'U' ou 0x55 : Déverrouillage ou Unlock Aig

D[0] = 'I' ou 0x49 : Réinitialisation Aig (Ex: Après une panne)

D[1] : Deux cas possibles (Droit ou bifurqué)

D[1] = 0x00 : Set Aiguille en position Droit

D[1] = 0x01 : Set Aiguille en position Bifurqué

D[2] : Activation ou non de l'auto verrouillage de l'aiguille (AutoLock)

D[2] = 'A' ou 0x41 : Mode Activation Autolock Aiguille. (Nécessitera une commande spéciale de déverrouillage de l'Aiguille avant un nouveau positionnement.)

D[2] != 'A' : L'aiguille ne sera pas verrouillée après son changement de position.

Datas en Mode « Compte rendu opérationnel ou CRO »

D[0] : Sera toujours égal à 'D' ou 0x44 pour Data Aiguille

D[1] : Quatre cas possibles:

D[1] = 0x00 : Aiguille en position Droit en mode « UnLocked »

D[1] = 0x01 : Aiguille en position Bifurque en mode « UnLocked »

D[1] = 0x80 : Aiguille en position Droit en mode « Locked »

D[1] = 0x81 : Aiguille en position Bifurque en mode « Locked »

D[2] : Compte rendu opérationnel ou CRO

D[2] = 0x00 : RAS, tout s'est bien passé

D[2] = 0x01 : Actionneur déjà dans état demandé

D[2] = 0x02 : Trame erronée

D[2] = 0x04 : Actionneur occupé, est déjà en mode « switcht »

D[2] = 0x05 : Le changement de position a été refusé, l'Aig est « Lock »

D[2] = 0x07 : Aiguille en panne.

Trames CAN pour la gestion des feux connectés à la carte Aiguille

On distingue deux types de trames:

- Une trame de positionnement d’une couleur à un feu. Un dispositif envoie une trame CAN à la carte « Actionneur » dont la D[0]=‘W’ et dont D[1] est la couleur désirée.
 - Une trame de compte rendu opérationnel renvoyée par la carte « Actionneur ». Dans ce cas D[0] = ‘D’, D[1] est la couleur du feu, et D[3] est le CRO. Cette même trame est aussi renvoyée en réponse à une « remote trame ».
- Rappel: Suivant l’état des switch, la carte peut piloter 0,1 ou 2 feux.

mc_InfraAIG_feu_A, mc_InfraAIG_feu_B

I/O

ID	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
(RC << 4) +0x4 : Feu A (RC << 4) +0x5 : Feu B	3	xx	xx	xx					

- D[0] : cdmc_InfraAIG_typeData_Feu_A ou cdmc_InfraAIG_typeData_Feu_B
D[0] = ‘W’ ou 0x57 : Trame d’action envoyée par un autre dispositif pour le positionnement d’une couleur à un feu.
D[0] = ‘D’ ou 0x44 : Trame de datas renvoyée par la carte Aiguille (Compte rendu opérationnel ou réponse à une « Remote trame »).
- D[1] : cdmc_InfraAIG_couleur_Feu_A ou cdmc_InfraAIG_couleur_Feu_B
0x00 : Feu éteint
0x01 : Feu rouge
0x02 : Feu vert
0x03 : Feu rouge clignotant
0x04 : Feu vert clignotant
0x05 : Feu orange
- D[2] : cdmc_InfraAIG_status_Feu_A ou cdmc_InfraAIG_status_Feu_B
0x00 : RAS
0x01 : MODULE_DEJA_DANS_ETAT_DEMANDE, Tout est OK, sauf que le feu était déjà dans la couleur demandée.
0x02 : La trame est erronée. Soit une erreur de DLC ou de datas non prévues

Trames CAN pour la gestion des capteurs « Hall » associés ou non aux aiguilles

Mode opératoire d’envoi de ces trames: Lorsqu’un capteur magnétique détecte une locomotive, une trame CAN est transmise. Aucune trame ne sera transmise lors du relâchement du capteur. Une nouvelle lecture de ce capteur aura lieu une seconde après, si la loco est toujours présente, une autre trame sera transmise et ceci, indéfiniment.

PS: Un capteur sera toujours associé à la carte, mais suivant la position du switch sur la carte Aiguille, on peut en attribuer un deuxième.

SW_ConfB --> OFF, Seul un capteur détection passage est activé
--> ON, Les deux capteurs sont activés

mc_InfraAIG_capteurPositionA, mc__InfraAIG_capteurPositionB

↓ IN

ID	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
(RC << 4) +0x6 : Capteur A (RC << 4) +0x7 : Capteur B	3	xx	xx	xx					

D[0] : cdmc_InfraAIG_typeData_CapteurA ou cdmc_InfraAIG_typeData_CapteurB.
D[0] ='D' ou 0x44 : Trame de datas qui indique l'état du capteur. Pas de remote trame de prévue.
Cette data n'est pas significative, c'est uniquement pour garder une certaine compatibilité avec des autres trames similaires.

D[1] : cdmc_InfraAIG_etat_CapteurA ou cdmc_InfraAIG_etat_CapteurB
0x01 : Présence loco au dessus du capteur

D[2] : cdmc_InfraAIG_status_CapteurA ou cdmc_InfraAIG_status_CapteurB
0x00 : En réponse à une remote trame, ou lors d'une première détection de la présence loco dessus le capteur
0x01 : La loco n'a pas quitté le capteur. Chaque seconde après la présence loco dessus le capteur, si la loco est toujours présente dessus le capteur, une trame est renvoyée avec cette valeur.

Trames CAN pour la gestion de l'alimentation en énergie des Tronçons et aiguilles

La carte Aiguille est connectée à une autre carte électronique de puissance qui gère l'alimentation en énergie des tronçons et des aiguilles. Une trame CAN de commande permet d'activer ou d'inhiber cette énergie.

Par défaut, chaque carte Aiguille validera tous les tronçons et aiguilles en mode « Alimentation Energie ».

PS: Est plutôt destiné aux étudiants pour TP qu'aux « Chercheurs ».

mc_InfraAIG_energieA, mc_InfraAIG_energieB

↓ IN

ID	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
(RC << 4) +0xA : Energie A (RC << 4) +0xB : Energie B	1	xx							

D[0] : cdmc_commande_EnergieA ou cdmc_commande_EnergieB
0x00: permet l'alimentation en énergie d'un tronçon ou d'aiguilles (Logique inverse)
0x01: Extinction de l'alimentation en énergie d'un tronçon ou d'aiguilles

Trames CAN pour la mémoire EEPROM de la carte Aiguille

mc_InfraAIG_EEPROM



I/O

ID	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
(RC << 4) + 0xE	x	xx	xx	xx	xx	xx	xx	xx	xx

D[0] : cdmc_InfraAIG_commande_EEPROM

‘O’ (Code ASCII de O = 0x4F)

-> Paramétrage OFFSET

‘W’ (Code ASCII de W = 0x57)

-> Ecriture EEPROM

‘D’ (Code ASCII de D = 0x44)

-> Lecture EEPROM (n'est pas une

commande, est uniquement renvoyé après une remote trame).

D[1] : cdmc_InfraAIG_data1_EEPROM

D[2] : cdmc_InfraAIG_data2_EEPROM

ou

D[1], D[2] : wdmc_InfraAIG_offset_EEPROM si D[0]='O'

D[3] : cdmc_InfraAIG_data3_EEPROM

D[4] : cdmc_InfraAIG_data4_EEPROM

D[5] : cdmc_InfraAIG_data5_EEPROM

D[6] : cdmc_InfraAIG_data6_EEPROM

D[7] : cdmc_InfraAIG_data7_EEPROM

NB: Pas d'auto-indentation de « wdmc_InfraAIG_offset_EEPROM »

ATTENTION, L'EEPROM peut être protégée en écriture à l'aide du Dipswitch sur la carte. Il suffit d'activer le switch EE->U vers la droite.

Si on tente une écriture dans l'EEPROM, un message warning est envoyé sur le bus avec les paramètres suivants:

Id warning = 0x0D : Trame dysfonctionnement survenu sur l'infrastructure.

D[0] -> Type de la carte qui a envoyée le message warning (0x01 -> carte balise)

D[1] -> Type warning, ici 0x10 pour type SOFTWARE

D[2] -> Valeur de la roue codeuse de la carte

D[3] -> Type du Warning soit 0xEE dans ce cas

PS: La led EEPROM du Bargraph s'allume aussi.

Mode d'emploi EEPROM:

- Paramétrage de l'adresse d'offset (pointeur):

Le PC envoie une trame standard avec D[0] = 'O' (Code ASCII de O = 0x4F), suivi de la valeur de l'offset codé de la façon suivante:

D[2] pour poids forts et D[1] pour poids faibles

OFFSET = (unsigned int) ((D[2] << 8) + D[1]) codage "Loco"

- Ecriture dans l'EEPROM :

Le PC envoie une trame standard avec D[0] = 'W' (Code ASCII de W = 0x57), suivi des valeurs à écrire dans l'EEPROM. Le DLC indique le nombre de valeurs à écrire.

A l'issu de l'écriture, la carte envoie un « ACK » pour signaler la fin de l'opération. Pour ce faire, une trame est renvoyée avec un DLC=1 et une D[0] = 'A' (0x41) pour ACK.

Si l'EEPROM était en mode « Read Only » (protection en écriture à l'aide du switch3 « bdmc_InfraAIG_modeProtect_EEPROM » de la carte Aiguille), la carte renverrait une trame « warning ».

PS: Si D[0] n'est ni 'O' ni 'W', la variable « cdmc_InfraAIG_cptTrameErronee » du message « mc_InfraAIG_statusDynamique » sera incrémenté. De même que le bit « bdmc_InfraAIG_def1TrameErronee » de cette trame. Par contre, le bit « bdmc_InfraAIG_def1TrameErronee » reste inchangé, on n'a pas eu d'erreur véritable au niveau de l'EEPROM.

- Lecture de l'EEPROM :

Le PC envoie une trame de Remote avec l'Id EEPROM

Le DLC de la trame remote indiquera le nombre de datas à renvoyer.

La carte renverra une trame avec

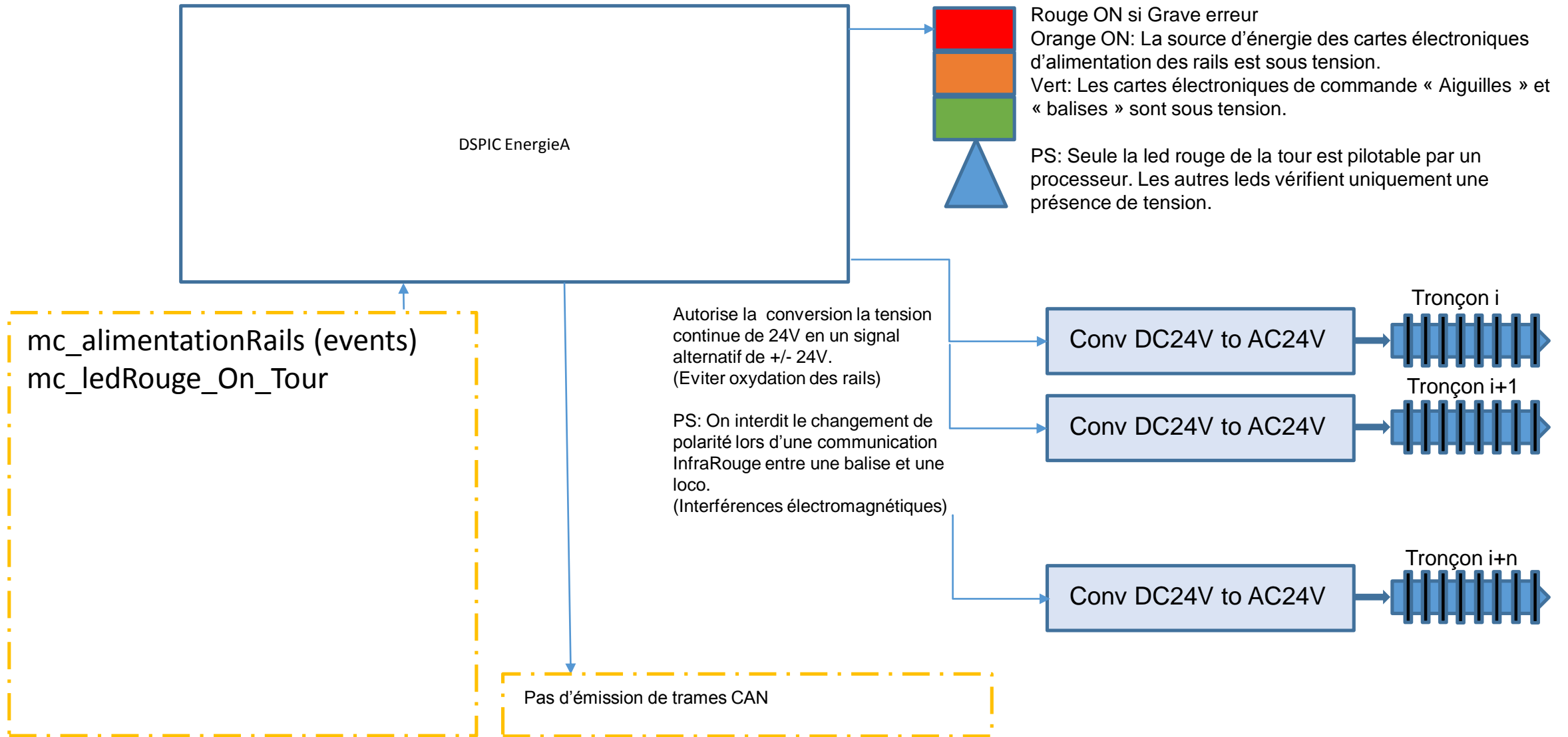
D [0] = 'D' (Code ASCII de D = 0x44), suivi du nombre de datas demandés.

NB: Le DLC max de la trame de remote est de 7, car D [0] = 'D',

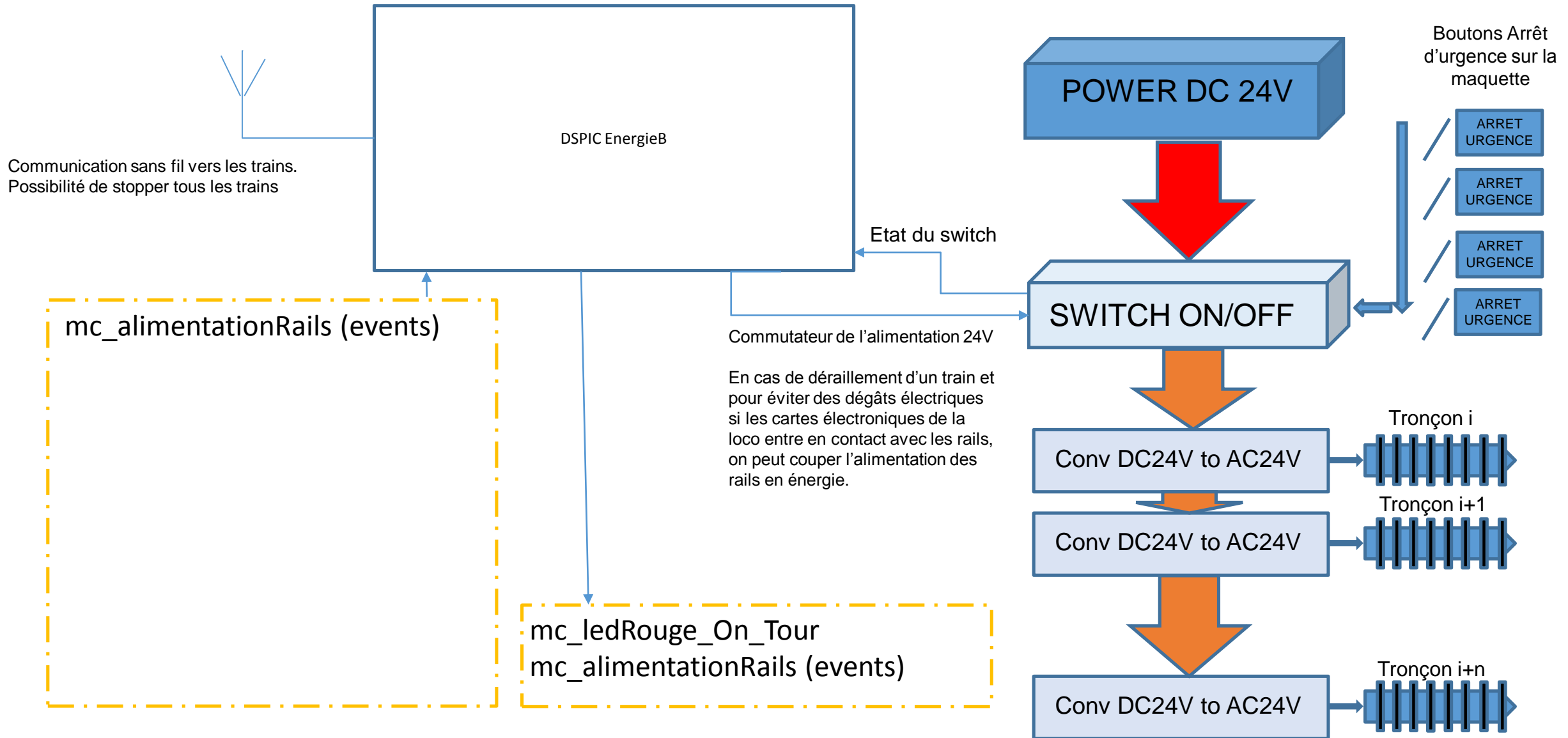
reste donc 7 octets de disponibles dans la trame de renvoi.

PS: Si DLC=8, alors les variables « cdmc_InfraAIG_cptTrameErronee » et

« bdmc_InfraAIG_def1TrameErronee » seront mis à jour.



Carte Infra EnergieB Sécurité Energie sur rails



Trames CAN accessibles aux différentes cartes de l'Infrastructure

mc_alimentationRails

ID	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
0xF0	1	xx							



OUT

L'envoi de cette trame permet de stopper le changement de polarité de la tension qui est appliquée sur les rails(+ ou – 24V)
Le changement de polarité provoque des interférences au niveau de la communication infrarouge. Son arrêt fiabilise la communication.
Le temps d'arrêt est fixé à 1 sec. Ensuite, sans autre trame de renvoi, le fonctionnement redevient « Normal ».

D[0] = cdmc_typeCommande_AlimentationRails

Bit7 de D[0] = Status ou CommandeBarre de la trame CAN

High: mode status (Trame envoyée par carte INFRA Energie B)
Low: mode commande (Trame à destination de la carte INFRA Energie B pour action)

Bit6, 5 et 4: Not used (low)

bit[3..0] = 0x1 (CAR_STOP_CHGT_POLARITY)

Arrêt du changement de polarité de la tension d'alimentation des rails

bit[3..0] = 0x2 (CAR_ENERGIE_SET_POWER_RAILS_OFF)

On coupe en énergie électrique l'alimentation des rails. Le switch est OFF
PS: l'alimentation 24V reste sous tension.

bit[3..0] = 0x4 (CAR_ENERGIE_SET_POWER_RAILS_ON)

Les rails sont alimentés en énergie électrique. Le switch est ON
Sert aussi à reprendre après un arrêt Urgence (Via trame CAN, Trame Xbee ou Boutons AR)
Par défaut, dans cet état, l'alimentation 24V est encore sous tension.

bit[3..0] = 0x8 (CAR_ENERGIE_OFF)

L'alimentation 24V n'est pas branchée ou a été éteinte manuellement par l'opérateur.

bit[3..0] = 0xF (CAR_PANIC_ENERGIE_OFF)

On coupe en énergie électrique l'alimentation des rails pour éviter des courts-circuits destructifs avec les trains qui auraient déraillés. On envoie aussi en liaison Wireless une commande à chaque train de mise hors tension instantanée.
PS: Pour quitter cet état, il faut renvoyer une trame CAN « Infra » avec le CAR_ENERGIE_SET_POWER_RAILS_ON

Trames CAN communes à l'Infrastructure

Trames CAN communes aux différentes cartes de l'Infrastructure
(Différentiation par l'Id CAN qui est une fonction des roues codeuses)

mc Infra crash

ID	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
0x0C	4	xx	xx	xx	xx				

OUT

```
D[0] = cdmc_InfraCrash_source
```

D[0] = 0x01 -> Carte balise

D[0] = 0x02 -> Carte aiguillage

D[0] = 0x03 -> Carte

D[0] = 0x04 -> Carte

```
D[1] = cdmc_InfraCrash_type
```

0x00: Hardware,

0x01: Software,

0x20: Causes externes

D[2] = cdmc_InfraCrash_RC

N° Carte soit la valeur des roues codeuses.

PS: si D[2] = 0x00, problème de lecture des roues codeuses.

On ne saura pas quelle carte incriminer avec cette info trame

CAN. Comme la carte est inutilisable et qu'il faut bien savoir

laquelle, il y aura clignotement des 2 leds bicolores

```
statusCom.
```

D[3] = cdmc_InfraCrash_code

Code particulier pour D[3] quand le bit 7 de D[3]= High

Bit0: NACK_MCP23008_RouesCodeuses.

Bit1: NACK_MCP23008_DIPSWITCH.

Bit2: NACK_MCP23008_BARGRAPH.

Bit3: NACK PCA9551D.

Bit4: NOT FOUND 24LC256

PS: il peut y avoir plusieurs bits High en même temps

[illegible]

Trames CAN communes aux différentes cartes de l'Infrastructure
(Différentiation par l'Id CAN qui est une fonction des roues codeuses)

mc Infra warning

(dysfonctionnement)

ID	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
0x0D	4	xx	xx	xx	xx				

↑
OUT

D[0] = cdmc_InfraWarning_source

D[0] = 0x01 -> Carte balise

D[1] = 0x02 -> Carte aiguillage

D[2] = 0x03 -> Carte

D[3] = 0x04 -> Carte

D[1] = cdmc_InfraWarning_type

0x00: Hardware,

0x10: Software,

0x20: Causes externes

0xFA: Fail communication BalA

0xFB: Fail communication BalB

Pour ces deux derniers cas, le codage de

D[3] est le suivant:

Bits[3..0]: compteur écrêté à 0xF du nombre

de dysfonctionnements intervenus lors de la communication IR. Le code n'a pas les « rattraper ».

Bit4: Erreur CKS

Bit5: Erreur TimeOUT

Bit6 et 7: Reserved à 0

D[2] = cdmc_InfraWarning_RC

N° Carte soit la valeur des roues codeuses.

D[3] = cdmc_InfraWarning_code

[illegible]

Trames CAN accessibles aux différentes cartes de l'Infrastructure

mc_ledRouge_On_Tour

ID	DLC	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
0x290	3	xx	xx	xx					



OUT

Un système de signalisation en forme de tour est installé au centre de la maquette train. Il comprends 3 systèmes d'affichage de leds.

- Vert: Indique que les cartes électroniques de la maquette sont sous tension. (N'est pas pilotable)
- Orange : Indique que les rails de la maquette sont sous tensions, On trouve une tension mode créneau de +24V à -24V. (N'est pas pilotable)
- Rouge : Indique un gros dysfonctionnement au niveau de la maquette qui nécessitera une intervention humaine. Cette led rouge est pilotable par une trame

CAN.

Le codage garde la même logique de codage que pour les actionneurs, c'est-à-dire que, après une trame de commande actionneur, une trame compte rendu est renvoyée par la carte actionneur.

D[0] = cdmc_typeData_LedRouge_On_Tour

D[0] = 'W' Pour signaler que c'est une commande d'écriture.

D[1] = 'D' En réponse à une remote trame

D[1] = cdmc_etat_LedRouge_On_Tour

0x00: Eteinte,

0x01: Allumée,

D[2] = cdmc_status_LedRouge_On_Tour

0x00: Don't care quand Envoie trame avec D[0] = 'W'

Valeur retournée après une trame d'écriture

0x00: RAS,

0x01: En réponse à une trame d'allumage ou d'extinction de la led, signifie que la led était déjà dans l'état demandé.

PS: On a une autre carte électronique, elle est aussi équipée d'un DsPIC qui gère l'état des leds qui équipent la tour « Leds » située au centre de la maquette.